Web Services Security Policy Language (WS-SecurityPolicy)

Version 1.0
December 18, 2002

Authors

Giovanni Della-Libera, Microsoft
Phillip Hallam-Baker, VeriSign
Maryann Hondo, IBM
Tomasz Janczuk, Microsoft
Chris Kaler, Microsoft
Hiroshi Maruyama, IBM
Anthony Nadalin (Editor), IBM
Nataraj Nagaratnam, IBM
Andrew Nash, RSA Security
Rob Philpott, RSA Security
Hemma Prafullchandra, VeriSign
John Shewchuk, Microsoft
Elliot Waingold, Microsoft
Riaz Zolfonoon, RSA Security

Copyright Notice

The WS-SecurityPolicy Specification may change before final release and you are cautioned against relying on the content of this specification.

Abstract

This document is an addendum to WS-Security and indicates the policy assertions for WS-Policy which apply to WS-Security.

Composable Architecture

By using the XML, SOAP and WSDL extensibility models, the WS* specifications are designed to be composed with each other to provide a rich Web services environment. WS-SecurityPolicy by itself does not provide a complete security solution for Web services. WS-SecurityPolicy is a building block that is used in conjunction with other Web service and application-specific protocols to accommodate a wide variety of security models.

Status

This WS-SecurityPolicy Specification is an initial public draft release and is provided for review and evaluation only. IBM, Microsoft, RSA Security, and VeriSign hope to solicit your contributions and suggestions in the near future. IBM, Microsoft, RSA Security, and VeriSign make no warrantees or representations regarding the specifications in any manner whatsoever.

Table of Contents

1. Introduction

Most Web service specifications indicate their associated policy assertions for use with WS-Policy. However, because WS-Security was published prior to WS-Policy, this addendum identifies these assertions.

2. Notations and Terminology

This section specifies the notations, namespaces, and terminology used in this specification.

2.1. Notational Conventions

The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC2119.

Namespace URIs (of the general form "some-URI") represents some application-dependent or context-dependent URI as defined in RFC2396.

WS-SecurityPolicy is designed to work with the general Web Services framework including WSDL service descriptions, UDDI businessServices and bindingTemplates and SOAP message structure and message processing model, and WS-SecurityPolicy should be applicable to any version of SOAP. The current SOAP 1.2 namespace URI is used herein to provide detailed examples, but there is no intention to limit the applicability of this specification to a single version of SOAP.

2.2. Namespaces

The XML namespace URI that MUST be used by implementations of this specification is:

```
        http://schemas.xmlsoap.org/ws/2002/12/secext
```

The following namespaces are used in this document:

Prefix	Namespace
S	http://schemas.xmlsoap.org/soap/envelope/
ds	http://www.w3.org/2000/09/xmldsig#
wsu	http://schemas.xmlsoap.org/ws/2002/07/utility
wsse	http://schemas.xmlsoap.org/ws/2002/12/secext
wsp	http://schemas.xmlsoap.org/ws/2002/12/policy
xsd	http://www.w3.org/2001/XMLSchema

2.3. Schema Files

The schema for this specification can be located at:

```
http://schemas.xmlsoap.org/ws/2002/12/secext
```

In this document reference is made to the *wsu:Id* attribute and the `<wsu:Created>` and `<wsu:Expires>` elements in a utility schema (http://schemas.xmlsoap.org/ws/2002/07/utility). The *wsu:Id* attribute and the <wsu:Created> and <wsu:Expires> elements were added to the utility schema with the intent that other specifications requiring such an ID or timestamp could reference it (as is done here).

2.4. Terminology

We introduce the following terms which are used throughout this document:

Policy – A *policy* is a set of domain-specific policy statements.

Policy Statement – A *policy statement* is a group of policy assertions.

Policy Assertion – A *policy assertion* represents an individual preference, requirement, capability or other property.

3. Security Extensions

The WS-Policy specification defines a framework for indicating a service's requirements and policies. In this section we define extensions to the <wsp:Policy> element for security properties for Web services. These extensions are primarily designed for describing policies related to the features defined in the WS-Security specification, but they can also be used for describing security requirements at a more abstract or transport-independent level.

To indicate support for WS-Security including the Addendum, the following policy assertion is used:

```
<wsp:SpecVersion wsp:Usage="wsp:Required"
          URI="http://schemas.xmlsoap.org/ws/2002/12/secext"/>
```

3.1. SecurityToken Assertion

A Web Service may require a requestor to attach a security token when it sends a request to the Web service. For example, a SAML authorization token issued by a trusted authorization authority needs to be presented to access sensitive data. Another example is that a binary security token containing an X.509 certificate needs to be presented for signing purpose.

The <wsse:SecurityToken> element is used to describe what security tokens are required and accepted by a Web service. It can also be used to express a Web Service's policy on security tokens that are included when the service sends out a message (e.g., as a reply message).

```
<SecurityToken wsp:Preference="..." wsp:Usage="..." >
  <TokenType>...</TokenType>
  <TokenIssuer>...</TokenIssuer>
  <Claims>...Token type-specific claims...</Claims>
  ...   (TokenType-specific details)
</SecurityToken>
```

The following describes the attributes and tags listed in the schema outlined above:

/SecurityToken

This identifies a security token assertion.

/SecurityToken/@wsp:Preference

This optional attribute specifies the preference of this particular alternative. The preference is expressed as an xsd:int. The higher the value of the preference, the greater the weighting of the expressed preference. If no preference is specified, a value of zero is assumed.

/SecurityToken/@wsp:Usage

This mandatory attribute indicates the usage of this assertion (e.g., required, optional, etc.) per WS-Policy.

/SecurityToken/TokenType

This mandatory element expresses the type of the security token for this assertion specified by a QName. This is extensible, but the following types are predefined:

QName	**Description**
wsse:X509v3	X.509 v3 certificate
wsse:Kerberosv5TGT	Kerberos V5 Ticket Granting Ticket
wsse:Kerberosv5ST	Kerberos V5 service ticket
wsse:UsernameToken	Username token defined in WS-Security
wsse:SAMLAssertion	SAML Assertion
wsse:XrMLLicence	XrML License

/SecurityToken/TokenIssuer

This optional element's contents are interpreted as the name of a trusted issuer (or names of trusted issuers).

/SecurityToken/Claims

This optional element contains data that is interpreted as describing type-specific claims that are expressed in the security token. TokenType-specific descriptions, such as required extensions in an X509 certificate, MUST be specified using this mechanism. Some of the TokenType-specific extensions are defined in Appendix I of this document.

/SecurityToken/{any}

This is a general extensibility mechanism to allow additional elements to be specified.

/SecurityToken/@{any}

This is an extensibility mechanism to allow additional attributes to be specified.

3.2. Integrity Assertion

Senders of messages can make use of the integrity mechanism defined in WS-Security to verify that specific aspects of the message have not been altered and to associate a security token (and associated claims) with those parts of the message. However, a service provider may require that specific portions of a message be signed and that specific algorithms and keys be used. For example, a service may require the body to be signed and only accept algorithms using SHA1 and an RSA key.

The WS-Policy operators, for example `<wsp:OneOrMore>`, can be used to specify different combinations of encryption and integrity assertions, or even choices of algorithms. All sub-elements use the same methodology as that described in WS-Policy. That is, there is an implicit <wsp:All> grouping.

If no algorithms are specified, then only the algorithms required by XML Signature are supported.

The <Integrity> element is used to indicate a required signature format.

It is possible to indicate a set of required claims for a signature that are independent of token type or authority, by specifying a <Claims> element within the <Integrity> assertion.

The schema outline for <Integrity>, an assertion about an integrity requirement, is as follows:

```
<Integrity wsp:Preference="..." wsp:Usage="...">
    <Algorithm Type="..." URI="..." wsp:Preference="..."/>
    <TokenInfo>
        <SecurityToken>...</SecurityToken>
    </TokenInfo>
    <Claims>...</Claims>
    <MessageParts Dialect="..." Signer="...">
        ...
    </MessageParts>
<Integrity>
```

The following describes the attributes and tags listed in the schema outlined above:

/Integrity
This identifies an integrity assertion.

/Integrity/@wsp:Preference
This optional attribute specifies the preference of this particular alternative. The preference is expressed as an xsd:int. The higher the value of the preference, the greater the weighting of the expressed preference. If no preference is specified, a value of zero is assumed.

/Integrity/@wsp:Usage
This mandatory attribute indicates the usage of this assertion (e.g., required, optional, etc.) per WS-Policy.

/Integrity/Algorithm
This optional element identifies an algorithm choice.

/Integrity/Algorithm/@Type
This optional attribute contains the type of the algorithm specified by a QName. This is extensible, but the following types are predefined:

QName	**Description**
wsse:AlgCanonicalization	Canonicalization
wsse:AlgSignature	Signature method
wsse:AlgTransform	Transformation

wsse:AlgDigest	Digest method

/Integrity/Algorithm/@URI

This optional attribute contains the URI reference of the algorithm.

/Integrity/Algorithm/@wsp:Preference

This optional attribute specifies the preference of this particular alternative. The preference is expressed as an xsd:int. The higher the value of the preference, the greater the weighting of the expressed preference. If no preference is specified, a value of zero is assumed.

/Integrity/Algorithm/{any}

The contents of this element are specific to the algorithm.

/Integrity/TokenInfo

This optional element identifies required security token formats. It should be noted that multiple key choices can be specified by using the `<wsp:OneOrMore>` operator specified in WS-Policy.

/Integrity/TokenInfo/SecurityToken

This optional element indicates a supported security token format or authority previously described.

/Integrity/TokenInfo/SecurityToken/@wsp:Preference

This optional attribute specifies the preference of this particular alternative. The preference is expressed as an xsd:int. The higher the value of the preference, the greater the weighting of the expressed preference. If no preference is specified, a value of zero is assumed.

/Integrity/TokenInfo/SecurityToken/{any}

This is an extensibility mechanism to allow different (extensible) types of security information to be specified.

/Integrity/TokenInfo/SecurityToken/@{any}

This is an extensibility mechanism to allow additional attributes to be specified.

/Integrity/Claims

This optional element contains data that is interpreted as describing general claims that must be expressed in the security token.

/Integrity/MessageParts

The contents of this element (of type xsd:string) is an expression that specifies the targets to be signed. The evaluation of the expression is determined by the optional dialect attribute. The default dialect is "http://www.w3.org/TR/1999/REC-xpath-19991116" indicating the expression is an XPath 1.0 expression. If there are multiple `<MessageParts>` elements specified, the concatenation specifies the parts are to be signed unless they are contained in a choice policy element (see WS-Policy). If the selection of the targets can be easily expressed using the "http://schemas.xmlsoap.org/2002/12/wsse#part" mechanism, then it is RECOMMENDED. Otherwise it is RECOMMENDED that a general XPath expression using the expressions defined in Appendix II be used.

/Integrity/MessageParts/@Dialect

The optional attribute identifies the expression dialect in use as a URI reference. If the attribute is not present, then XPath 1.0 is assumed.

URI	Meaning

http://www.w3.org/TR/1999/REC-xpath-19991116 (Default)	An XPath 1.0 location path that identifies the nodes to be protected. The XPath expression is evaluated against the S:Envelope element node to select which nodes are to be protected. Additionally, the expression SHOULD use the functions defined in Appendix I of WS-PolicyAssertions (where appropriate).
http://schemas.xmlsoap.org/2002/12/wsse#part	A list of message parts to be protected that are identified using the set of pre-defined functions defined in Appendix II of WS-PolicyAssertions. The functions are is evaluated against the S:Envelope element node.

/Integrity/MessageParts/@Signer

This optional attribute contains a list of one or more URI references that indicate which nodes must provide a signature. The pre-defined values are:

URI	**Description**
http://schemas.xmlsoap.org/2002/12/secext/originalSender (default)	The originator of the message (at a minimum) must sign the identified element(s).

/Integrity/MessageParts/@{any}

This extensibility allows for additional attributes to be specified.

/Integrity/{any}

This is an extensibility mechanism to allow additional elements to be specified.

/Integrity/@{any}

This is an extensibility mechanism to allow additional attributes to be specified.

The following example illustrates the use of this declaration. In this example, only messages with signatures using Exclusive Canonicalization, signed using RSA-SHA1 with an X.509 security token will be processed. The signature must cover the entire body as well as a header block with the specified element name.

```
    <wsse:Integrity wsp:Usage="wsp:Required">
        <wsse:Algorithm  Type="wsse:AlgCanonicalization"
                 URI="http://www.w3.org/Signature/Drafts/xml-exc-c14n"/>
        <wsse:Algorithm Type="wsse:AlgSignature"
                       URI=" http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
        <wsse:SecurityToken>
            <wsse:TokenType>wsse:X509v3</wsse:TokenType>
        </wsse:SecurityToken>
        <MessageParts
```

```
             Dialect="http://schemas.xmlsoap.org/2002/12/wsse#soap">
        S:Body some-URI:HeaderBlockElementName
    </MessageParts>
</wsse:Integrity>
```

3.3. Confidentiality Assertion

Senders of messages can make use of the confidentiality mechanism defined in WS-Security to ensure that specific aspects of the message are encrypted. However, a requestor may require that specific portions of a message be encrypted and that a specific algorithm be used. For example, a service may require the body to be encrypted using triple-DES.

The WS-Policy mechanisms, for example <wsp:OneOrMore>, can be used to specify different confidentiality combinations or confidentiality and integrity combinations, or even choices of algorithms.

If no algorithms are specified, then only the algorithms required by XML Encryption are supported.

The <Confidentiality> element is used to indicate a required encryption format.

The schema outline for this element is as follows:

```
<Confidentiality wsp:Preference="..." wsp:Usage="...">
    <Algorithm Type="..." URI="..." wsp:Preference="..."/>
    <KeyInfo>
        <SecurityToken .../>
        <SecurityTokenReference .../>
        ...
    </KeyInfo>
    <MessageParts Dialect="...">
        ...
    </MessageParts>
</Confidentiality>
```

The following describes the attributes and tags listed in the schema outlined above:

/Confidentiality

This identifies the encryption format for XML Encryption.

/Confidentiality/@wsp:Preference

This optional attribute specifies the preference of this particular alternative. The preference is expressed as an xsd:int. The higher the value of the preference, the greater the weighting of the expressed preference. If no preference is specified, a value of zero is assumed.

/Confidentiality/@wsp:Usage

This mandatory attribute indicates the usage of this assertion (e.g., required, optional, etc.) per WS-Policy.

/Confidentiality/Algorithm

This optional element identifies an algorithm choice using the same element defined for signature-related algorithms. For encryption we pre-define the following algorithm types:

QName	**Description**
wsse:AlgEncryption	Encryption

/Confidentiality/KeyInfo

This optional element identifies required key formats. Note that different key options can be specified using a `<wsp:OneOrMore>` operator.

/Confidentiality/KeyInfo/SecurityToken

This optional element identifies a required key as described in the security token requirement section above. Refer to the `<Integrity>` assertion for details on this element.

/Confidentiality/KeyInfo/SecurityTokenReference

This optional element identifies a security token that should be used for the encryption.

/Confidentiality/KeyInfo/{any}

This extensibility mechanism permits a security token to be specified which should be used for the encryption.

/Confidentiality/MessageParts

The contents of this element (of type xsd:string) is an expression that specifies the targets to be encrypted. The evaluation of the expression is determined by the optional dialect attribute. The default dialect is "http://www.w3.org/TR/1999/REC-xpath-19991116" indicating the expression is an XPath 1.0 expression. If there are multiple `<MessageParts>` elements specified, the concatenation specifies the parts are to be encrypted unless they are contained in a choice policy element (see WS-Policy). If the selection of the targets can be easily expressed using the "http://schemas.xmlsoap.org/2002/12/wsse#part" mechanism, then it is RECOMMENDED. Otherwise it is RECOMMENDED that a general XPath expression using the expressions defined in Appendix II be used.

/Confidentiality/MessageParts/@Dialect

The optional attribute identifies the expression dialect in use as a URI reference. If the attribute is not present, then XPath 1.0 is assumed.

URI	**Meaning**
http://www.w3.org/TR/1999/REC-xpath-19991116 (Default)	An XPath 1.0 location path that identifies the nodes to be encrypted. The XPath expression is evaluated against the S:Envelope element node. Additionally, the expression SHOULD use the functions defined in Appendix I of WS-PolicyAssertions (where appropriate).
http://schemas.xmlsoap.org/2002/12/wsse#part	A list of message parts to be encrypted that are identified using the set of pre-

	defined functions defined in Appendix II of WS-PolicyAssertions. The functions are is evaluated against the S:Envelope element node.

/Confidentiality/MessageParts/@{any}

This extensibility allows for additional attributes to be specified.

/Confidentiality/@{any}

This is an extensibility mechanism to allow additional attributes to be specified.

The following example illustrates the use of this declaration. In this example, the body must be encrypted using triple-DES.

```
<wsse:Confidentiality wsp:Usage="wsp:Required">
    <wsse:Algorithm Type="wsse:AlgEncryption"
                    URI="http://www.w3.org/2001/04/xmlenc#3des-cbc"/>
    <MessageParts>
        wsp:GetInfosetForNode(wsp:GetBody(.))
    </MessageParts>
</wsse:Confidentiality>
```

The following example illustrates the use of confidentiality and integrity. In this example, the body is required to be encrypted prior to being signed.

```
<wsse:Confidentiality wsp:Usage="wsp:Required">
    <wsse:Algorithm Type="wsse:AlgEncryption"
                    URI="http://www.w3.org/2001/04/xmlenc#3des-cbc"/>
    <MessageParts>
        wsp:GetInfosetForNode(wsp:GetBody(.))
    </MessageParts>
</wsse:Confidentiality>

<wsse:Integrity wsp:Usage="wsp:Required">
    <wsse:Algorithm  Type="wsse:AlgCanonicalization"
             URI="http://www.w3.org/Signature/Drafts/xml-exc-c14n"/>
    <wsse:Algorithm Type="wsse:AlgSignature"
                   URI=" http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
    <wsse:SecurityToken>
        <wsse:TokenType>wsse:X509v3</wsse:TokenType>
    </wsse:SecurityToken>
    <MessageParts
            Dialect="http://schemas.xmlsoap.org/2002/12/wsse#soap">
        S:Body
    </MessageParts>
</wsse:Integrity>
```

3.4. Visibility Assertion

Some intermediaries may require that parts of the message be visible to them. That is, they either need to be passed in the clear (unencrypted), or there must be an encryption binding for the intermediary.

The <Visibility> element is used to indicate portions of a message that MUST be able to be processed by an intermediary or endpoint.

The schema outline for this element is as follows:

```
    <Visibility wsp:Usage="...">
        <MessageParts Dialect="...">
            ...
        </MessageParts>
    </Visibility>
```

The following describes the attributes and tags listed in the schema outlined above:

/Visibility

This identifies the portions of a message that must be "visible" to an actor.

/Visibility/@wsp:Usage

This mandatory attribute indicates the usage of this assertion (e.g., required, optional, etc.) per WS-Policy.

/Visibility/MessageParts

The contents of this element (of type xsd:string) is an expression that specifies the targets to be visible. The evaluation of the expression is determined by the optional dialect attribute. The default dialect is "http://www.w3.org/TR/1999/REC-xpath-19991116" indicating the expression is an XPath 1.0 expression. If there are multiple <MessageParts> elements specified, the concatenation specifies the parts are to be visible unless they are contained in a choice policy element (see WS-Policy). If the selection of the targets can be easily expressed using the "http://schemas.xmlsoap.org/2002/12/wsse#part" mechanism, then it is RECOMMENDED. Otherwise it is RECOMMENDED that a general XPath expressions using the expressions defined in Appendix II be used.

/Visibility/MessageParts/@Dialect

The optional attribute identifies the expression dialect in use as a URI reference. If the attribute is not present, then XPath 1.0 is assumed.

URI	**Meaning**
http://www.w3.org/TR/1999/REC-xpath-19991116 (Default)	An XPath 1.0 location path that identifies the nodes to be visible. The XPath expression is evaluated against the S:Envelope element node. Additionally, the expression SHOULD use the functions defined in Appendix I of

	WS-PolicyAssertions (where appropriate).
http://schemas.xmlsoap.org/2002/12/wsse#part	A list of message parts to be visible that are identified using the set of pre-defined functions defined in Appendix II of WS-PolicyAssertions. The functions are is evaluated against the S:Envelope element node.

/Visibility/MessageParts/@{any}

This extensibility allows for additional attributes to be specified.

/Visibility/{any}

This is an extensibility mechanism to allow additional elements to be specified.

/Visibility/@{any}

This is an extensibility mechanism to allow additional attributes to be specified.

The following example illustrates the use of this declaration. In this example, the body must be visible to the http://www.fabrikam123.com endpoint.

```
<wsse:Visibility wsp:Usage="wsp:Required">
    <MessageParts>
        wsp:GetInfosetForNode(wsp:GetBody(.))
    </MessageParts>
</wsse:Visibility>
```

3.5. Security Header Assertion

The <Security> header as defined in WS-Security provides several degrees of freedom. In this section we provide a policy statement to constrain certain aspects of this header.

The schema outline for this element is as follows:

```
<SecurityHeader MustPrepend="..."
                MustManifestEncryption="..."
                wsp:Usage="..."/>
```

The following describes the attributes and tags listed in the schema outlined above:

/SecurityHeader

This identifies specific behaviors when using the <Security> header.

/SecurityHeader/@MustPrepend

This optional attribute, if true, indicates that entries to the <Security> header MUST be prepended. If false (the default), then entries are NOT REQUIRED to be pre-pended.

/SecurityHeader/@MustManifestEncryption

This optional attribute, if true, indicates that only encryptions listed or referenced from the <Security> header will be processed; any encryptions in the message not referenced will be ignored. If false (the default), then the processor MUST search the message for applicable encryptions to process.

/SecurityHeader/@wsp:Usage

This mandatory attribute indicates the usage of this assertion (e.g., required, optional, etc.) per WS-Policy.

/SecurityHeader/@{any}

This extensibility allows for additional attributes to be specified.

/SecurityHeader/{any}

This is an extensibility mechanism to allow additional elements to be specified.

The following example illustrates the use of this declaration. In this example, senders MUST prepend entries in the <Security> header and MUST list or reference encryptions that need to be processed.

```
<wsse:SecurityHeader wsp:Usage="wsp:Required"
                     MustPrepend="true"
                     MustManifestEncryption="true"/>
```

3.6. MessageAge Assertion

The <wsse:MessageAge> element is used to indicate the recipients acceptable time period before messages are declared "stale" and discarded (based on creation times as defined in WS-Extensions). If a policy specifies a <wsse:MessageAge> element, then the service that is the target of such policy requires the <Timestamp> header (from the WS-Security specification) in the received message to evaluate and enforce the policy. If a message is received without timestamp information and a <wsse:MessageAge> element is present in the policy, a service MAY discard the message, but is not required to do so.

The schema outline for <wsse:MessageAge> is as follows:

```
<wsse:MessageAge wsp:Usage="..." wsp:Preference="..." Age=.../>
```

The following describes the attributes and tags listed in the schema outlined above:

/MessageAge

This specifies the maximum age for message.

/MessageAge/@wsp:Usage

This mandatory attribute indicates the usage of this assertion (e.g., required, optional, etc.) per WS-Policy.

/MessageAge/@wsp:Preference

This optional attribute specifies the preference of this particular alternative. The preference is expressed as an xsd:int. The higher the value of the preference, the greater the weighting of the expressed preference. If no preference is specified, a value of zero is assumed.

/MessageAge/@Age

This required attribute specifies the actual maximum age timeout for a message expressed in seconds.

/MessageAge/@{any}

This is an extensibility mechanism to allow additional attributes to be specified.

The following example illustrates the use of this declaration for one hour expiration:

```
<wsse:MessageAge wsse:Usage="wsp:Required" Age="3600"/>
```

4. Security Considerations

It is strongly RECOMMENDED that policies and assertions be signed to prevent tampering.

It is RECOMMENED that policies SHOULD NOT be accepted unless they are signed and have an associated security token to specify the signer has proper claims for the given policy. That is, a party shouldn't rely on a policy unless the policy is signed and presented with sufficient claims.

It should be noted that the mechanisms described in this document could be secured as part of a SOAP message using WS-Security or embedded within other objects using object-specific security mechanisms.

5. Acknowledgements

We would like to thank the following people for their contributions towards this specification:

Erik Christensen, Microsoft
Slava Kavsan, RSA Security
Scott Konersmann, Microsoft
David Melgar, IBM
John Linn, RSA Security
Steve Millet , Microsoft
Keith Stobie, Microsoft
Kent Tamura, IBM

6. References

[KEYWORDS]
S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels," RFC 2119, Harvard University, March 1997

[RFC2068]
IETF Standard, "Hypertext Transfer Protocol -- HTTP/1.1" January 1997

[SOAP]
W3C Note, "SOAP: Simple Object Access Protocol 1.1," 08 May 2000.

[URI]
T. Berners-Lee, R. Fielding, L. Masinter, "Uniform Resource Identifiers (URI): Generic Syntax," RFC 2396, MIT/LCS, U.C. Irvine, Xerox Corporation, August 1998.

[WS-Policy]
"Web Services Policy Framework", BEA, IBM, Microsoft, SAP, December 2002

[WS-PolicyAssertion]
"Web Services Policy Assertions Language", BEA, IBM, Microsoft, SAP, December 2002

[WS-PolicyAttachment]
"Web Services Policy Attachment Language", BEA, IBM, Microsoft, SAP, December 2002

[WS-Security]
"Web Services Security Language", IBM, Microsoft, VeriSign, April 2002.
"WS-Security Addendum", IBM, Microsoft, VeriSign, August 2002.

"WS-Security XML Tokens", IBM, Microsoft, VeriSign, August 2002.

[WSDL]
"Web Services Description Language", IBM/Microsoft, 15 March 2001.

[XML-C14N]
W3C Recommendation, "Canonical XML Version 1.0," 15 March 2001.

[XML-Encrypt]
W3C Recommendation, "XML Encryption Syntax and Processing," 10 December 2002.

[XML-ns]
W3C Recommendation, "Namespaces in XML," 14 January 1999.

[XML-Schema1]
W3C Recommendation, "XML Schema Part 1: Structures,"2 May 2001.

[XML-Schema2]
W3C Recommendation, "XML Schema Part 2: Datatypes," 2 May 2001.

[XML Signature]
W3C Recommendation, "XML Signature Syntax and Processing," 12 February 2002.

Appendix I – TokenType-Specific Elements

This appendix defines TokenType specific extensions to the <SecurityToken> assertions.

I.1. X.509v3 Token

When the TokenType is wsse:X509v3, the optional <TokenIssuer> element in a <SecurityToken> assertion, if present, MUST be the distinguished name of the certificate issuer, either the actual issuing CA or the root CA who vouches the issuing CA. As well, special claim elements are outlined in the following schema.

```
<SecurityToken wsp:Preference="..." wsp:Usage="..." wsu:id="...">
  <TokenType>wsse:X509v3</TokenType>
  <TokenIssuer>...</TokenIssuer>
  <Claims>
      <SubjectName MatchType="...">...</SubjectName>
      <X509Extension OID="..." Critical="..." MatchType="...">
        ...
      </X509Extension>
  </Claims>
</SecurityToken>
```

The following describes the attributes and tags listed in the schema outlined above:

/SecurityToken/Claims/SubjectName
This optional element specifies the requirements on the subject name of the X.509 certificate. The string value of this element MUST match the subject name of the certificate.

/SecurityToken/Claims/SubjectName/@MatchType

The value of this optional attribute MAY be one of wsse:Exact or wsse:Prefix. The interpretation of the matching operation is given in the table below. If this attribute is omitted, the default value is wsse:Prefix.

QName	**Description**
wsse:Exact	The values must be exactly the same.
wsse:Prefix (default)	The specified value must be the prefix of the value in the certificate

/SecurityToken/Claims/X509Extension/

This optional element specifies the requirements on the extensions of the X.509 certificate. The string value of this element must match the value of the extension.

/SecurityToken/Claims/X509Extension/@OID

The value of this mandatory attribute MUST be a string representation of OID of this extension.

/SecurityToken/Claims/X509Extension/@Critical

The value of this optional attribute is of type Boolean. If the value is true, it indicates that the specified extension in the certificate must be critical. If the value is false, it indicates that the specified extension in the certificate must not be critical. If this attribute is omitted, no requirement is given on whether the specified extension is critical or not.

/SecurityToken/Claims/X509Extension/@MatchType

The value of this optional attribute MUST be one of wsse:Exact or wsse:Prefix. The interpretation of the matching operation is given in the table above. If this attribute is omitted, the default value is wsse:Prefix.

QName	**Description**
wsse:Exact	The values must be exactly the same.
wsse:Prefix (default)	The specified value must be the prefix of the value in the certificate

I.2. Kerberos Token

When either the TokenType is either wsse:Kerberosv5TGT or wsse:Kerberosv5ST, the optional <TokenIssuer> element in a <SecurityToken> assertion, if present, MUST identify the Kerberos realm. Additional extensions to the <SecurityToken> element are outlined in the following schema.

```
<SecurityToken wsp:Preference="..." wsp:Usage="..." wsu:Id="...">
  <TokenType>wsse:Kerberosv5TGT</TokenType>
  <TokenIssuer>...</TokenIssuer>
  <Claims>
    <SubjectName MatchType="...">...</SubjectName>
    <ServiceName>...</ServiceName>
</Claims>
</SecurityToken>
```

The following describes the attributes and tags listed in the schema outlined above:

/SecurityToken/Claims/SubjectName

This optional element specifies the requirements on the subject of the Kerberos ticket. The string value of this element must match the client's PrincipalName (the cname field the Ticket defined in RFC-1510)..

/SecurityToken/Claims/SubjectName/@MatchType

The value of this optional attribute MAY be one of wsse:Exact or wsse:Prefix. The interpretation of the matching operation is given in the table below. If this attribute is omitted, the default value is wsse:Prefix.

QName	**Description**
wsse:Exact	The values must be exactly the same.
wsse:Prefix (default)	The specified value must be the prefix of the value in the ticket

/SecurityToken/Claims/ServiceName

The string value of this mandatory element is the service's PrincipalName (the sname field of the ticket defined in RFC-1510).

I.3. Username Token

When the TokenType is wsse:UsernameToken, the TokenIssuer element in a SecurityToken assertion MUST be absent. Additional extensions to the SecurityToken element are outlined in the following schema

```
<SecurityToken wsp:Preference="..." wsp:Usage="..." wsu:id="...">
  <TokenType>wsse:UsernameToken</TokenType>
  <Claims>
    <SubjectName MatchType="...">...</SubjectName>
    <UsePassword wsp:Usage="..." Type="..."/>
  </Claims>
</SecurityToken>
```

The following describes the attributes and tags listed in the schema outlined above:

/SecurityToken/Claims/SubjectName

This optional element specifies the requirements on the contents of the <Username> element of the <UsernameToken>. The string value of this element must match the string value of the <Username> element.

/SecurityToken/Claims/SubjectName/@MatchType

The value of this optional attribute MAY be one of wsse:Exact, wsse:Prefix, and wsse:Regexp. The interpretation of the matching operation is given in the table below. If this attribute is omitted, the default value is wsse:Prefix.

QName	**Description**
wsse:Exact	The values must be exactly the same.
wsse:Prefix (default)	The specified value must be the prefix of the value in the certificate
wsse:Regexp	The specified value is an regular expression that matches the value in the

	token

/SecurityToken/Claims/UsePassword/

This optional element specifies the requirements on the <Password> element in the <UsernameToken>.

/SecurityToken/Claims/UsePassword/@wsp:Usage

This mandatory attribute indicates the usage of the <Password> element (e.g., required, optional, etc.) as defined in WS-Policy. If the usage is wsp:Rejected then the specified type is not supported. If no type is specified, then the policy does not allow passwords to be passed.

/SecurityToken/Claims/UsePassword/@Type

The value of this optional attribute MAY be of type one of the values given in the following table (as is defined in WS-Security). If this attribute is omitted, then any type of password MAY be specified.

QName	**Description**
wsse:PasswordText	The <Type> attribute of this <UserNameToken> is <wsse:PasswordText> (i.e., plain text password is used). This is the default value.
wsse:PasswordDigest	The Type attribute of this <UsernameToken> is <wsse:PasswordDigest> (i.e., digested password is used).